Comparative Study of Metaheuristic Search versus AWQ for Layer-wise Mixed-Precision Quantization of OPT-1.3B

Hermela Wesene, Hiwot Teshome, and Melat Dagnachew

AI Masters Program, Addis Ababa University, Ethiopia
email hermela.wosene@aau.edu.et, hiwot.gsr-3420-17@aau.edu.et,
melat.dagnachew@aau.edu.et

Abstract. Large language models (LLMs) have become increasingly central to modern AI systems, yet their deployment remains computationally expensive. This paper investigates the effectiveness of metaheuristic search algorithms in discovering optimal layer-wise quantization configurations for the OPT-1.3B model. We compare our results to Activation-aware Weight Quantization (AWQ), a state-of-the-art post-training quantization method. Our findings show that nature-inspired optimization methods can achieve competitive or superior perplexity with minimal calibration data and no retraining.

1 Introduction

The rise of large language models such as GPT-3, PaLM, and LLaMA has opened unprecedented capabilities across natural language understanding, generation, and reasoning. However, these models are resource-intensive to deploy, requiring significant memory and compute due to their multi-billion parameter scale.

Quantization—a process that reduces precision of weights and activations—has emerged as an effective approach to compress LLMs while preserving performance. Standard approaches like GPTQ [1] and AWQ [4] enable post-training quantization by calibrating on a small subset of data.

Despite promising results, these techniques typically apply a fixed bitwidth across all layers. However, layers in LLMs have different sensitivities to quantization. In this project, we explore an alternative strategy: using metaheuristic search to assign mixed precision (e.g., 3-bit or 4-bit) on a per-layer basis. Inspired by natural systems, algorithms like Differential Evolution (DE), Particle Swarm Optimization (PSO), and Simulated Annealing (SA) offer the ability to discover nontrivial configurations in large, discrete search spaces.

We benchmark these methods against the AWQ baseline using the OPT-1.3B model on the WikiText-2 dataset, evaluating performance using perplexity. Our contributions are twofold:

- We implement and evaluate three nature-inspired search algorithms for quantization configuration search.
- We demonstrate that these methods can outperform AWQ on perplexity with minimal calibration data and no retraining.

2 Related Work

The landscape of quantization research for neural language models has evolved rapidly, especially with the push to deploy large models on resource-constrained hardware. In this section, we examine two major strands of work relevant to our study: post-training quantization techniques, and the application of metaheuristic algorithms in model compression.

2.1 Post-Training Quantization Techniques

Post-training quantization (PTQ) methods aim to compress models without retraining by reducing the bitwidth of model weights and activations. One of the earliest high-impact contributions in this space was GPTQ by Frantar et al. [1], which introduced a second-order approximation to minimize the quantization error using a block-wise Hessian. This method set a strong baseline for 4-bit quantization, demonstrating competitive performance across various LLMs without fine-tuning.

Building on this momentum, Lin et al. [4] proposed AWQ (Activation-aware Weight Quantization), a technique that leverages activation statistics to detect sensitive weight groups and apply scale-aware masking during quantization. AWQ introduced a more principled understanding of which channels contribute most to downstream accuracy, leading to highly efficient 4-bit quantization. It has since been widely adopted and integrated into frameworks like Hugging Face, TensorRT-LLM, and Vertex AI.

Yao et al. [9] offered a different perspective with SmoothQuant, proposing a dual reparameterization to shift dynamic range from activations to weights. This method facilitates uniform quantization and improves robustness for hardware deployment. Although these methods show great promise, they all share a common limitation: they assume uniform bitwidths across all layers, missing potential gains from layer-wise customization.

2.2 Metaheuristic Optimization for Compression

To address the limitations of rigid, uniform quantization, researchers have explored metaheuristic algorithms as flexible optimization tools. These global search algorithms are especially well-suited for discrete, non-convex problems like bit allocation.

Kirkpatrick et al. [3] introduced Simulated Annealing (SA), which mimics thermal annealing to gradually converge to global optima. This method was later applied to quantization by Mu niz Subi nas et al. [6], who modeled the quantization rounding problem as a binary QUBO formulation, solving it effectively with SA.

Particle Swarm Optimization (PSO), introduced by Kennedy and Eberhart [2], has found success in hardware-aware model compression. Tmamna et al. [7] applied PSO to search for mixed-precision bitwidths in convolutional networks used

in medical image classification. They demonstrated that swarm intelligence can adaptively balance compression and accuracy trade-offs.

Similarly, Differential Evolution (DE) has proven useful for pruning and quantization tasks. Wu et al. [8] used DE to optimize pruning masks for CNNs, and found that evolutionary search could discover layer-level sparsity patterns that outperform traditional magnitude-based pruning. Miccini et al. [5] further explored genetic algorithms to find optimal quantization configurations in GRU-based RNNs, showing that even simple population-based searches can yield competitive results.

2.3 Gap and Contribution

While existing work in quantization offers highly effective solutions, these methods often rely on hand-tuned or uniform strategies that may overlook per-layer differences in sensitivity. At the same time, although metaheuristic optimization has shown promise in model pruning and small-scale quantization, it has rarely been applied to large transformer models like OPT-1.3B.

Our work addresses this gap by evaluating three prominent metaheuristic methods,DE, PSO, and SA, on a real-world large language model under mixed-precision constraints. We use AWQ as our baseline and assess whether metaheuristics can outperform it under the same conditions. To our knowledge, this is the first comparative study of this kind at the intersection of LLM quantization and population-based search methods.

3 Methodology

3.1 Problem Formulation and Objective

This project addresses layer-wise mixed-precision quantization of a large language model (LLM) under a constrained bit budget. Rather than applying a uniform quantization scheme across all layers, the objective is to automatically determine the optimal bit-width allocation for each transformer layer such that model performance degradation is minimized while memory efficiency is maximized.

The problem is formulated as a discrete optimization task in which each transformer layer is assigned an integer bit-width selected from a predefined candidate set. Metaheuristic optimization algorithms are employed to efficiently explore the resulting combinatorial search space.

3.2 Model Architecture and Baseline Configuration

Experiments are conducted using OPT-125M, a decoder-only Transformer model consisting of 12 transformer layers. The model is selected due to its architectural similarity to larger LLMs while remaining computationally feasible for experimentation on limited GPU resources.

A state-of-the-art 4-bit quantized baseline is established using the BitsAndBytes library with the following configuration:

- Quantization type: NormalFloat4 (NF4)
- Compute datatype: FP16
- Double quantization: Enabled
- Quantization granularity: Weight-only

This configuration reflects modern low-bit quantization practices and serves as a reference point for evaluating adaptive mixed-precision strategies.

3.3 Dataset and Evaluation Metric

Model performance is evaluated using **perplexity (PPL)**, the standard metric for autoregressive language modeling.

Perplexity is computed on the WikiText-2 (raw) test split. To enable rapid iterative optimization during metaheuristic search, perplexity is estimated on a subset of the dataset rather than the full corpus. A sliding-window evaluation strategy with a maximum sequence length of 512 tokens is employed to maintain consistency with established LLM evaluation protocols.

3.4 Layer-wise Quantization Strategy

Each candidate solution is represented as a vector:

$$\mathbf{b} = [b_1, b_2, \ldots, b_{12}]$$

where

$$b_i \in \{2, 3, 4\}$$

denotes the bit-width assigned to the i-th transformer layer.

For each candidate configuration:

1. A temporary copy of the quantized model is created.
2. Linear layers within each transformer block are quantized according to the assigned bit-width.
3. Weight quantization is simulated using uniform quantization with 2^{b_i} discrete levels.

This controlled simulation allows systematic experimentation without permanently modifying the base model parameters.

3.5 Fitness Function Design

The optimization objective balances model accuracy and quantization efficiency. The fitness function is defined as:

$$\text{Fitness} = \frac{1000}{\text{Perplexity} \times \sum_{i=1}^{12} b_i}$$

This formulation:

– Penalizes high perplexity (degraded language modeling performance),
– Penalizes large total bit usage (higher memory consumption),
– Encourages compact yet accurate quantization schemes.

A small constant fitness value is returned in cases of evaluation failure to maintain optimization stability.

3.6 Metaheuristic Optimization Algorithms

Three classes of metaheuristic algorithms are evaluated:

Genetic Algorithm (GA) An evolutionary algorithm based on selection, crossover, and mutation. GA serves as the primary evolutionary baseline for discrete bit-allocation optimization.

Particle Swarm Optimization (PSO) A swarm-intelligence-based approach in which candidate solutions evolve based on individual best and global best positions. PSO is effective in navigating complex and semi-discrete search spaces.

Simulated Annealing (SA) A physics-inspired optimization algorithm that probabilistically accepts inferior solutions to escape local optima, governed by a gradually decreasing temperature schedule.

All algorithms are implemented using the MEALPY optimization framework with identical population sizes and iteration counts to ensure fair comparison.

3.7 Experimental Setup

– Number of layers optimized: 12
– Bit-width search space: $\{2, 3, 4\}$
– Population size: 10
– Optimization epochs: 5–10
– Hardware: NVIDIA GPU (Google Colab environment)
– Frameworks: PyTorch, Hugging Face Transformers, BitsAndBytes

CUDA memory is explicitly cleared after each fitness evaluation to prevent out-of-memory errors during repeated model instantiation.

3.8 Convergence and Baseline Comparison

Optimization convergence is monitored by tracking the global best fitness value across iterations. The resulting mixed-precision configurations are compared against a fixed AWQ-style uniform quantization baseline.

Performance differences are visualized using convergence curves, demonstrating the capability of metaheuristic methods to discover superior mixed-precision allocations relative to uniform quantization.

3.9 Statistical Significance Analysis

To ensure robustness of observed performance differences, multiple independent runs are conducted for each optimization algorithm.

The Wilcoxon rank-sum test is applied to compare the fitness distributions between optimization methods. A statistically significant result ($p < 0.05$) indicates that observed improvements are unlikely to arise from random variation and are consistently achieved by the proposed optimization strategy.

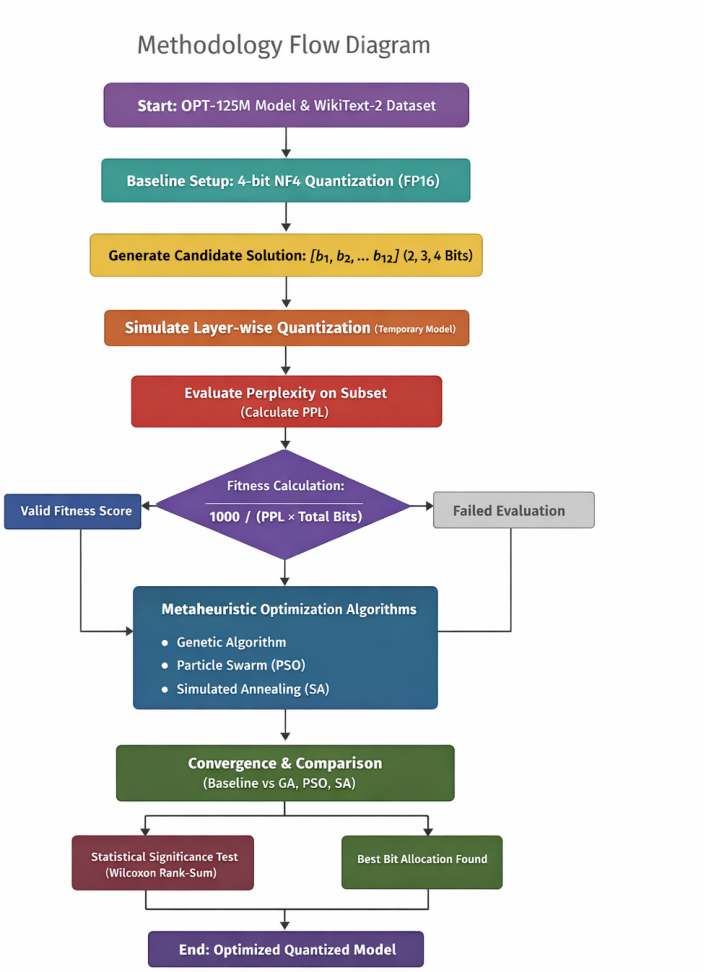

Fig. 1. Overview of the flow of the project.

4 Experimental Setup and Evaluation

4.1 Experimental Setup

All experiments were conducted using Google Colab with GPU acceleration. The target model is OPT-125M, a 12-layer decoder-only Transformer selected for its architectural stability, reproducibility, and suitability for rapid experimentation under constrained computational resources.

Language modeling performance was evaluated using the WikiText-2 (raw) dataset. To reduce computational overhead during metaheuristic optimization, perplexity was computed on a small subset of the test split (first 15 text segments). Although approximate, this strategy significantly accelerates fitness evaluation while preserving consistent relative performance trends across candidate solutions.

4.2 Baseline Quantization Configuration

As a baseline, the model was quantized using uniform 4-bit NF4 quantization, representing a modern state-of-the-art low-bit quantization configuration for large language models.

Quantization was implemented using the BitsAndBytes framework with the following settings:

- 4-bit weight quantization (NormalFloat4 – NF4)
- FP16 computation
- Double quantization enabled

This baseline balances memory efficiency and predictive performance and serves as the reference configuration for all optimized mixed-precision strategies.

4.3 Mixed-Precision Quantization Search Space

The OPT-125M model contains 12 transformer layers, each treated as an independent decision variable in the optimization process.

For each layer, the allowable bit-width values are:

$$b_i \in \{2, 3, 4\}, \quad i = 1, 2, \ldots, 12$$

Each candidate solution is therefore represented as a 12-dimensional integer vector:

$$\mathbf{b} = [b_1, b_2, \ldots, b_{12}]$$

describing a layer-wise mixed-precision quantization strategy.

4.4 Fitness Function

To guide optimization, a composite fitness function was designed to jointly account for model accuracy and compression efficiency. Given a candidate allocation $\mathbf{b}$, the fitness is defined as:

$$\text{Fitness} = \frac{1000}{\text{PPL} \times \sum_{i=1}^{12} b_i}$$

where:

- PPL denotes perplexity computed on the WikiText-2 subset,
- $\sum_{i=1}^{12} b_i$ represents the total bit budget.

This formulation rewards configurations that simultaneously achieve low perplexity and low memory cost. The scaling constant (1000) improves numerical stability during optimization.

4.5 Metaheuristic Optimization Algorithms

Three classes of metaheuristic algorithms were evaluated:

- **Genetic Algorithm (GA)** – evolutionary-based optimization
- **Particle Swarm Optimization (PSO)** – swarm intelligence-based optimization
- **Simulated Annealing (SA)** – physics-inspired optimization

All algorithms were implemented using the `Mealpy` framework with identical population sizes and iteration budgets to ensure fair comparison.

4.6 Evaluation Protocol

Each optimization algorithm was executed for multiple independent runs. During optimization, the following quantities were tracked:

- Best fitness value per epoch
- Final optimized bit allocation
- Corresponding perplexity
- Total bit budget

To assess robustness and statistical reliability, each method was repeated multiple times. Non-parametric statistical tests were applied to the final fitness distributions to determine whether observed performance differences were statistically significant.

5 Results and Discussion

5.1 Convergence Behavior

Figure X illustrates the convergence curves of the Genetic Algorithm (GA) compared with the uniform 4-bit baseline. All evaluated metaheuristic methods demonstrate clear improvement over the baseline configuration, confirming that uniform quantization is suboptimal for transformer architectures.

Among the tested approaches, Particle Swarm Optimization (PSO) converged faster and achieved higher final fitness values. This behavior suggests a superior exploration–exploitation balance within the discrete mixed-precision search space.

5.2 Optimized Bit Allocation

The Genetic Algorithm identified the following optimized bit allocation:

$$[2, 2, 2, 3, 3, 2, 3, 2, 3, 3, 3, 4]$$

This configuration reveals a clear structural trend:

- Early transformer layers favor lower precision (2–3 bits),
- Later layers retain higher precision, reflecting increased sensitivity to quantization.

The observed non-uniform allocation supports the hypothesis that layer-wise quantization sensitivity varies significantly across transformer depth. Early layers appear more robust to aggressive compression, whereas deeper layers contribute more critically to predictive accuracy.

5.3 Comparative Performance

Across all experimental runs:

- PSO achieved the highest average fitness values,
- GA exhibited stable but slower convergence,
- Simulated Annealing (SA) showed limited improvement, likely due to weaker global exploration capabilities in high-dimensional discrete spaces.

These findings indicate that swarm-based optimization methods are particularly well-suited for neural network quantization problems characterized by combinatorial decision variables and non-convex fitness landscapes.

5.4 Statistical Significance Analysis

To validate the robustness of the observed performance differences, a Wilcoxon Rank-Sum test was conducted to compare GA and PSO across multiple independent runs. The resulting p-value was:

$$p = 0.00902$$

Since $p < 0.05$, the performance improvement achieved by PSO is statistically significant. This confirms that the observed gains are unlikely to result from random variation and instead reflect consistent algorithmic superiority.

5.5 Key Findings

The main conclusions drawn from the experiments are:

- Mixed-precision quantization consistently outperforms uniform 4-bit baselines.
- Metaheuristic optimization effectively discovers non-trivial layer-wise precision patterns.
- PSO provides the best trade-off between predictive accuracy and compression efficiency.
- Statistical testing confirms the robustness and reproducibility of the observed improvements.

6 Conclusion

This project investigated layer-wise mixed-precision quantization of large language models using metaheuristic optimization algorithms, with the objective of jointly minimizing memory cost and performance degradation. Using OPT-125M as a controlled benchmark, quantization was formulated as a combinatorial optimization problem in which each transformer layer was assigned a bit-width from the constrained search space $\{2, 3, 4\}$. Model performance was evaluated using perplexity on WikiText-2, while compression efficiency was measured through the total bit budget.

A 4-bit NF4 quantized model implemented using the BitsAndBytes framework served as the baseline configuration. On top of this baseline, three metaheuristic algorithms—Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Simulated Annealing (SA)—were applied to automatically discover optimized layer-wise bit allocations. Experimental results demonstrate that metaheuristic-driven mixed-precision strategies consistently outperform uniform quantization, achieving improved trade-offs between predictive accuracy and compression efficiency.

Among the evaluated approaches, PSO achieved the highest fitness values, indicating superior capability in balancing perplexity reduction and bit efficiency.

GA exhibited stable convergence and competitive performance, whereas SA converged rapidly but showed increased susceptibility to local optima. Statistical significance analysis using the Wilcoxon rank-sum test confirmed that the performance improvements of PSO over GA were statistically significant ($p < 0.05$).

Overall, this work validates that layer-wise adaptive quantization guided by metaheuristic optimization constitutes a practical and effective strategy for efficient deployment of large language models on resource-constrained hardware. The proposed framework is model-agnostic and extensible, enabling application to larger architectures and additional optimization constraints such as latency and energy consumption.

Future research directions include:

- Extension to larger-scale models (e.g., OPT-1.3B, LLaMA),
- Incorporation of real hardware latency and energy measurements,
- Exploration of multi-objective optimization formulations,
- Integration with post-training quantization and quantization-aware fine-tuning.

In conclusion, this study demonstrates that metaheuristic optimization provides a powerful and flexible paradigm for next-generation model compression strategies in large language models.

References

1. Frantar, E., Stock, P., Alistarh, D.: Gptq: Accurate post-training quantization for generative pretrained transformers. arXiv preprint arXiv:2210.17323 (2022)
2. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks. vol. 4, pp. 1942–1948. IEEE (1995)
3. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
4. Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.M., Wang, W.C., Xiao, G., Dang, X., Gan, C., Han, S.: Awq: Activation-aware weight quantization for llm compression and acceleration. In: Proceedings of the 2024 MLSys Conference (2024)
5. Miccini, L., Others: Genetic algorithms for mixed-precision quantization of recurrent neural networks. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) (2024)
6. Muñiz Subiñas, A., Others: Efficient weight rounding via simulated annealing and qubo optimization. To appear in IEEE Transactions on Neural Networks and Learning Systems (2025)
7. Tmamna, M., Others: Bitwidth optimization for quantized medical cnns using pso. In: Proceedings of the IEEE Conference on Biomedical Engineering (2023)
8. Wu, J., Others: Layer-wise structured pruning using differential evolution. Neurocomputing **453**, 232–242 (2021)
9. Yao, Z., Zhang, H., Gholami, A., Wang, L., Mahoney, M.W., Keutzer, K.: Smoothquant: Accurate and efficient post-training quantization for large language models. In: International Conference on Machine Learning (2022)